REMARKS

In an Office Action dated March 16, 2006, the Examiner rejected claims 1, 5-7, 10-11, 15-17, 19-20, 26-28, 30-31 and 34-35 under 35 U.S.C. 102(b) as anticipated by Witt (US 5,623,627); and rejected claims 2-4, 8-9, 12-14, 21, 23-25, 92 and 32-33 under 35 U.S.C. 103(a) as unpatentable over *Witt* in view of Lane (US 2004/0225859).

Applicants have amended the independent claims herein to more specifically define the scope of the present invention. In particular, the claims have been amended to clarify that a set of addresses which map to a single associativity set in the first cache are distributed among multiple associativity sets in the second cache, and vice versa. As amended the claims are patentable over the cited art.

Applicant's invention is intended to improve the ratio of cache hits where multi-level set-associative caches are used by spreading the effect of certain "hot" associativity sets. Set-associative caches are well known in the art. For reasons which are complex and not always fully understood, where a set-associative cache is used, some of the sets may tend to run "hot", i.e., receive a relatively large proportion of data access references. Cache lines in these "hot" sets are evicted from the cache sooner, and have a greater probability of being needed again.

In a conventional multi-level cache design, an associativity set in a higher level cache maps directly to one or more associativity sets in a lower level cache. In some cases, where the lower level cache is smaller (e.g., victim caches), multiple associativity sets in a higher level cache map directly to a single associativity set in the lower level cache. The ratio of associativity sets is almost always a power of two. For example, a higher level cache may have 256 associativity sets and a lower level cache may have 512 associativity sets. In such a case, each associativity set of

the higher level cache maps directly to two respective associativity sets of the lower level cache, so that each set of the lower level cache gets exactly ½ of the addresses from a set of the higher level cache mapped into it.

The problem with these architectures is that they tend to propagate the hot sets into the lower level cache. I.e., if a given set in the higher level cache is hot, and there is a one-to-one mapping of associativity sets, then the corresponding set of the lower level cache will also be hot, reducing the ability of the lower level cache to mitigate hot sets in the higher level cache. Even where each of the higher level cache's sets map to multiple sets of the lower level cache, this problem frequently arises. Because the multiple sets are almost always powers of two, it frequently happens that most of the references in the hot set in the higher level cache will map to a single set of the lower level cache, rather than be distributed equally. This problem is particularly acute in the case of a victim cache, which is usually smaller than the higher level cache from which it receives evicted cache lines.

Applicant addresses this problem by using different address mappings for the higher and lower level caches, such that each set of the higher level cache is distributed among multiple sets of the lower level cache, and vice versa. This achieves a much greater randomization of the memory accesses in the two caches, so that if there is a hot set in one cache, it is not likely to correspond to any single hot set in the other cache. Preferably, the associativity sets are grouped in congruence groups, which are groups of associativity sets in both cache levels sharing a portion of the same address decode mapping, and within a congruence group, the addresses of each set in the higher level are distributed among all the sets of the lower level more or less equally (and vice versa). The use of congruence groups simplifies address decode logic, but it is not strictly required. Preferably, either the number of sets in the higher level in a congruence group, or the number in the lower level, is something other than a power of two, which achieves greater randomization.

Witt, cited by the Examiner, discloses a two level arrangement, in which a higher level cache is divided into data and instruction cache portions, and a larger lower level cache ("replacement cache") acts similarly to a victim cache. *Witt* discloses that the two caches are preferably set-associative, but are indexed using different addresses. In particular, *Witt* neither teaches nor suggests a multi-level cache architecture, in which addresses which map to one set in the higher level cache map to multiple sets in the lower level cache, and vice versa.

Applicant's representative claim 1, as amended, recites:

> 1. A digital data processing device, comprising:
> at least one processor;
> a memory;
> a first cache for temporarily holding portions of said memory, said first cache containing a plurality of addressable associativity sets, each associativity set containing one or more respective cache lines and corresponding to a respective first cache subset of a plurality of discrete first cache subsets of addresses for accessing said first cache; and
> a second cache for temporarily holding portions of said memory, said second cache containing a plurality of addressable associativity sets, each associativity set containing one or more respective cache lines and corresponding to a respective second cache subset of a plurality of discrete second cache subsets of addresses for accessing said second cache;
> wherein each said associativity set of said first cache and each said associativity set of said second cache is contained in a respective congruence group of a plurality of congruence groups, each congruence group containing a respective plurality of associativity sets of said first cache and a respective plurality of associativity sets of said second cache;
> wherein *addresses of the first cache subset corresponding to each respective associativity set of said first cache are allocated among each of the plurality of second cache subsets* corresponding to respective associativity sets in said second cache within the same congruence group as the respective associativity set of said first cache. [emphasis added]

The remaining independent claims are not identical in scope and recite the key feature in different terms (not always including the limitation of a congruence group), but the basic feature of one-to-multiple mapping of associativity sets is recited in all independent claims.

The final clause in the claim above contains the key limitation, and should be considered carefully. It states: "wherein addresses of the first cache subset corresponding to ***each respective associativity set...***" [i.e., ***for each and every associativity set*** in the first cache, the addresses which map to it] "...are allocated ***among each of the plurality of*** second cache subsets..." [i.e., are distributed among all the second cache subsets, leaving none out, so that each of the second cache subsets gets some of the addresses] "...corresponding to respective associativity sets in said second cache within the same congruence group as the respective associativity set of said first cache" [i.e., the associativity sets being in the same congruence group]. Since the previous clause recites that each congruence group contains multiple associativity sets of the first cache and multiple sets of the second cache, there are multiple first cache associativity sets mapping to each associativity set of the second cache, and vice versa.

This key limitation is simply not taught or suggested by *Witt*. *Witt* discloses a multilevel cache architecture, in which the first level cache is divided into an instruction and a data cache, and the second level is a common replacement or victim cache shared by both caches of the first level. Address bits 11:4 are used to index an associativity set of the first level cache, and bits 12:4 are used to index and associativity set of the second level cache.[1] Therefore addresses of each associativity set of the first level are mapped to one of two associativity sets in the second level (distributed evenly among the two sets), depending on the value of address bit 12. But the reverse is not true, nor is there any suggestion whatsoever in *Witt* that it should be made so. I.e., for any associativity set of the second level, its addresses map to one and only one set of the first level. Therefore, the applicable claim limitation discussed above is not met. There is nothing in *Witt*

---

[1] *Witt* discloses that bits 11:4 of the linear address (a form of virtual address which may require translation) are used to index an associativity set in the first level cache, and bits 12:4 of the physical address are used to index an associativity set in the second level cache. However, bits 11:0 of the linear address are always the same as bits 11:0 of the physical address. See Col 4, lines 25-39.

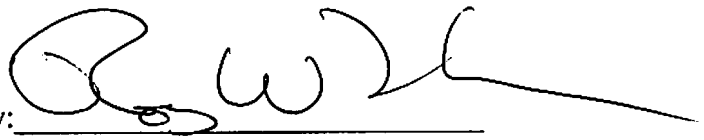
which would suggest altering his architecture or cache indexing scheme so that there are one-to-multiple mappings of associativity sets in both directions, as claimed by applicant.

Lane, the secondary reference, discloses a hashing of address bits to map into a set associative cache, but does not disclose other aspects of the multi-cache architecture claimed by applicant. It likewise does not teach or suggest, alone or in combination with *Witt*, the key features of applicant's invention.

In view of the foregoing, applicant submits that the claims are now in condition for allowance, and respectfully requests reconsideration and allowance of all claims. In addition, the Examiner is encouraged to contact applicant's attorney by telephone if there are outstanding issues left to be resolved to place this case in condition for allowance.

Respectfully submitted,

AARON C. SAWDEY

By: 

Roy W. Truelson
Registration No. 34,265

Telephone: (507) 202-8725